Premiers programmes, fonctionnels ou interactifs

Exercice 1 - Un peu d'arithmétique

On rappelle que les commandes `//` et `%` nous permettent d'obtenir le quotient et le reste de la division euclidienne par un entier.

1. Dans le langage Python, construire le programme *diviseurs* qui pour tout entier n non nul donné, teste si les entiers $k \in [\![1, n]\!]$ sont des diviseurs de n, puis renvoie la liste des diviseurs de n.

2. Que renvoie l'instruction : `len(diviseurs(24))` ?

3. Un nombre entier est alors dit **premier** s'il possède exactement deux diviseurs : 1 et lui-même.
   Construire la fonction booléenne *premier* qui pour tout entier n non nul, teste si celui-ci est premier ou non.

4. Construire le programme *valuation* qui pour tout couple (p, n) donné, renvoie la puissance avec laquelle p intervient dans n de sorte que :

```python
In : valuation(5,24); valuation(3,24); valuation(2,24)

0
1
3
```

5. En déduire le programme *decomposition* qui pour tout entier n donné, supérieur ou égal à 2, renvoie la décomposition de n en produit de facteurs premiers.
   On veillera à renvoyer le résultat sous la forme d'une liste s'écrivant :
   $$[p_1, \alpha_1, p_2, \alpha_2, \ldots, p_k, \alpha_k]$$
   où p_i désignent des nombres premiers et α_i les valuations associées dans n.
   Pour finir, on pourra aussi ajouter un test conditionnel sur l'argument afin de renvoyer un message d'erreur si $n < 2$.

6. Tester votre dernier programme pour $n \in \{7632, 1000204, 2^{29} - 1\}$.

Exercice 2 - Moyenne de vos colles

On cherche ici à construire un programme intéractif qui nous permet de calculer la moyenne des notes de colles d'un élève.

Dans le langage Python, construire le programme *moyenne* qui ne prend pas d'argument, mais demande à l'utilisateur le nombre n de notes, puis lui demande de rentrer ses n notes les unes après les autres.
Le programme affiche alors un graphe représentant l'évolution des notes de colles, puis renvoie la moyenne de ces n notes.

Exercice 3 - Encore un jeu de calcul mental

On souhaite construire un jeu de calcul mental dans lequel :

- le programme nous demande un niveau de difficulté $k \in \{1, 2, 3, 4, 5\}$
- puis l'ordinateur propose cinq additions de deux nombres choisis au hasard entre 1 et $k * 100$.

1. Dans le langage Python, construire le programme *jeu* qui ne prend pas d'argument, puis exécute les instructions précédentes avant de renvoyer le score obtenu. On pourra ajouter un compteur pour déterminer le temps de calcul du joueur.

2. Adapter votre programme afin de pouvoir faire jouer deux joueurs l'un après l'autre. Le programme commencera par demander le nom des joueurs, puis renverra à la fin le nom du gagnant et les scores associés.

Exercice 4 - Calcul intégral

Soit f une fonction continue sur $[a, b]$, alors on rappelle que :

$$\int_a^b f(t)\, dt = F(b) - F(a)$$

avec F une primitive de f sur $[a, b]$.

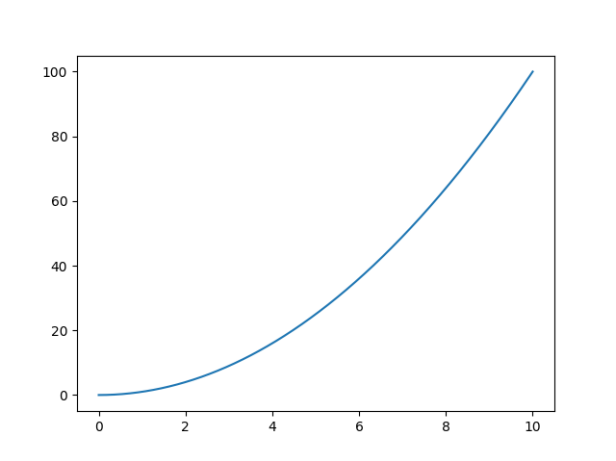
1. Soient $a, b \in \mathbb{R}_+^*$. Donner une primitive de $t \longmapsto t^n$ sur $[a, b]$.

2. Dans le langage Python, construire alors le programme *integral* qui pour tout triplet (a, b, n) donné, renvoie la valeur de $\int_a^b t^n\, dt$.

3. On importe les fonctions intégrées au sous-module **scipy.integrate**.
   Que renvoie l'instruction `quad(lambda t:t**2,1,2)` ?

4. En déduire la fonction *approx* qui pour tout triplet (a, b, n) donné, renvoie une valeur approchée de $\int_a^b t^n\, dt$.

5. On fixe $n = 2$. En utilisant vos souvenirs de terminale, construire de la même façon un programme vous permettant de calculer une autre approximation de $\int_a^b t^2\ dt$?



C'est d'ailleurs sur ce genre de méthode numérique que s'appuie la fonction `quad`*...*

Exercice 5 - Résolution approchée d'une équation de la forme $f(x) = 0$

On considère la fonction polynôme définie sur $\mathbb{R}$ par $f(x) = x^3 + x + 1$.

1. Justifier que l'équation $f(x) = 0$ possède une unique solution $\alpha \in \mathbb{R}$.
2. Dans le langage Python, définir la fonction f puis vérifier que $f(-5)f(5) < 0$.
3. Construire le programme *racineentiere* qui ne prend pas d'argument, mais teste si les entiers $k \in [\![-5, 5]\!]$ sont solutions de l'équation $f(x) = 0$.
4. Construire alors la fonction *approximation* qui pour tout $\epsilon > 0$ donné, détermine un encadrement de α tel que :

$$a \leq \alpha \leq b \text{ avec } b - a < \epsilon$$